Doc Code: AP.PRE.REQ

PRE-APPEAL BRIEF REQUEST FOR REVIEW	Docket Number: SVL920030045US1	
I hereby certify that this correspondence is being transmitted via the EFS-Web System to the USPTO on: May 8, 2007 Signature: /Janaki K. Davda/ Typed or Printed Name: Janaki K. Davda Reg. No. 40,684	**Application Number:** 10/675,265	**Filed:** September 29, 2003
	First Named Inventor: K.. BROWN et al.	
	Art Unit: 2167	**Examiner:** Robert M. Timblin

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached five (5) sheet(s).
Note: No more than five (5) pages may be provided.

I am the:

☐ applicant/inventor

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

☒ attorney or agent of record.
Registration Number Registration No. 40,684

☐ attorney or agent acting under 37 CFR 1.34
Registration number if acting under 37 CFR 1.34

/Janaki K. Davda/
Signature

Janaki K. Davda
Typed or Printed Name

(310) 553-7973
Telephone Number

May 8, 2007
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required*.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	K. BROWN et al.	Examiner	Robert M. Timblin
Serial No.	10/675,265	Group Art Unit	2167
Filed	September 29, 2003	Docket No.	SVL920030045US1
TITLE	METHOD, SYSTEM, AND PROGRAM FOR PREDICATE PROCESSING BY ITERATOR FUNCTIONS		

CERTIFICATE UNDER 37 CFR 1.8:
I hereby certify that this correspondence is being transmitted through the USPTO EFS-Web system over the Internet to Robert M. Timblin of the U.S. Patent and Trademark Office on May 8, 2007.

__/Janaki K. Davda/_____
Janaki K. Davda

PRE-APPEAL BRIEF REQUEST FOR REVIEW ARGUMENTS

Dear Examiner Timblin:

Applicants request a pre-appeal brief review of the Examiner's rejection of claims 1-39, which stand rejected 35 U.S.C. 102(e) as being anticipated by Andrei et al. (U.S. Patent No. 6,801,905 B2) in a Final Office Action mailed February 8, 2007.

Anticipation requires that the identical invention must be shown in a single reference in as complete detail as is contained in the claims. Richardson v. Suzuki Motor Co., 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

In An Advisory Action mailed on April 17, 2007, the Examiner states: "Andrei teaches operators module that contains optimization time descriptions of those runtime iterators which are used for evaluation during the optimization process (co. 24, lines 29-34). The Examiner respectfully submits that mentioning "runtime iterators" at least suggests invoking an iterator function (i.e., iterators executed or invoked at runtime)." Applicants respectfully submit that a *suggestion* of invoking iterators does not show the identical invention of *when an iterator function included in a statement is invoked.*

In the Advisory Action, the Examiner further states: "Andrei further describes iterators in a query execution plan which is a tree of relational algorithms applied to input tuple streams (col. 6, line 23-40). As the runtime iterator is used for evaluation the claimed limitation of invoking an iterator function is taught." Applicants respectfully submit that iterators in a query

execution plan do not show the identical invention of *an iterator function included in a statement*.

In the Advisory Action, the Examiner also states: "Further, Andrei also teaches invoking access methods (col. 11, lines 28-32 and col. 15, line 30-37). Andrei teaches invoking several times an enforcer rule (col. 15, line 34-35) while operators that are used for optimizing runtime iterators contain, in part, a description of grouping enforcement (co. 24, lines 25-35). Therefore, Andrei teaches when an iterator function is invoked." Applicants respectfully submit that access methods and the enforcer rule do not show the identical invention of *an iterator function included in a statement*.

As described in Applicants' Background of the Invention, a conventional iterator function receives a set of arguments and returns a table to the SQL statement that invokes the function (e.g., Specification, page 1, paragraph 4). In particular, the iterator function creates a virtual table with a result set, and then the qualification (i.e., predicate) is applied to the virtual table to filter rows of data in the virtual table (e.g., Specification, page 3, paragraph 9). For example, an application program submitting SQL statement (2) is interested in articles with a score (assigned by the text_search() iterator function) greater than or equal to 0.9, but, because the text_search() iterator function is not aware of the score qualification, the text_search() iterator function returns a virtual table with all articles that have "Bush" in the same sentence as "recession," without applying the predicate for the score (e.g., Specification, page 3, paragraphs 10-11).

To avoid this problem, claims 1, 14, and 27 describe, under control of an iterator function processor that executes as part of a data store engine, when an iterator function included in a statement is invoked, obtaining one or more predicates included in the statement and applying the one or more predicates to a row of data.

Applicants request review because the Examiner has not cited any part of the Andrei patent that discloses the identical invention claimed of obtaining one or more predicates included in the statement, applying the one or more predicates to a row of data, if applying the one or more predicates results in a match, returning the row of data, and, if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match occurs *"under control of an iterator function processor that executes as part of a data store engine, when an iterator function included in a statement is invoked*.

With reference to "when an iterator function included in a statement is invoked", the Examiner cites figure 8A, Col. 11, lines 4-13, and Col. 24, lines 25-67. Figure 8A describes receiving a query that is parsed and normalized and describes adding subplans to an equivalence class. Applicants respectfully submit that Figure 8A does not show the identical invention of "when an iterator function included in a statement is invoked". Col. 11, lines 4-13, describe that SQL statements are passed to the parser which converts the statements into a query tree. Applicants respectfully submit that this does not show the identical invention of "when an iterator function included in a statement is invoked".

At Col. 6, lines 23-27, the Andrei patent describes that a query execution plan or QEP is a demand-driven tuple stream "iterator" tree, which is a data structure that is interpreted by the relational database management system's execution module or engine. Col. 24, lines 25-67, describe that the physical operators module implements optimization time physical operators that are used for the optimization of the Volcano runtime iterators and describes that they are not themselves runtime iterators that implement the open/next/close protocol, that, rather, the physical operators module contains optimization time descriptions of those runtime iterators which are used for evaluation and costing during the optimization process. Applicants respectfully submit that mention of runtime iterators does not show the identical invention of when an iterator function included in a statement is invoked.

Also, the Andrei patent describes at Col. 11, lines 33-40, that, operating under the control of these instructions, the execution unit generates calls into lower-level routines, such as the access methods, for retrieving relevant information (e.g., row) from the database table, and, after the plan has been executed by the execution unit, the server returns a query result or answer table back to the client(s). Applicants respectfully submit that merely retrieving relevant information does not show the identical invention of "when an iterator function included in a statement is invoked . . . if applying the one or more predicates results in a match, returning the row of data, and, if applying the one or more predicates does not result in a match, searching for another row of data for which application of the one or more predicates results in a match."

Applicants request review because the Examiner has not cited any part of the Andrei patent that discloses the identical invention claimed in claims 6, 19, and 32 of, under control of a data store engine, invoking the iterator function one or more times, until receiving a done

indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine.

The Examiner cites figures 5 and 8B and Col. 11, lines 33-40, as teaching invoking the iterator function one or more times, until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine. Applicants respectfully submit that figure 5 merely mentions "applied predicates", while figure 8B describes determining a final plan for execution of the query and generating a physical operator tree. Col. 11, lines 33-40 describe that, operating under the control of these instructions, the execution unit generates calls into lower-level routines, such as the access methods, for retrieving relevant information (e.g., row) from the database table, and, after the plan has been executed by the execution unit, the server returns a query result or answer table back to the client(s). Applicants respectfully submit that the identical invention of invoking the iterator function one or more times until receiving a done indicator from the iterator function, wherein each invocation of the iterator function results in receiving either a row of data for which at least one predicate has been applied or the done indicator and wherein one or more of the predicates included in the statement that have not been applied by the iterator function are applied by the data store engine is not shown by the Andrei patent.

With the Andrei patent, the execution unit 369 generates calls into lower-level routines, such as the access methods 370, for retrieving relevant information (e.g., row 355) from the database table 350, and, after the plan has been executed by the execution unit, the server returns a query result or answer table back to the client(s) (Col. 11, lines 35-40). Thus, with the Andrei patent the execution unit itself executes the plan. Thus, the Andrei patent teaches away from one or more of the predicates included in the statement that have not been applied by the iterator function being applied by the data store engine.

Also, Applicants respectfully submit that the Examiner has not cited any part of the Andrei patent that discloses the identical invention of receiving a statement including an iterator function and one or more predicates. Instead, the Andrei patent describes runtime iterators

which are used for evaluation and costing during the optimization process (Col. 24, lines 31-32) and an iterator tree which is a data structure (Col. 6, lines 24-25).

Claims 12, 25, and 38 describe the interaction between a data store engine and an iterator function processor in processing an iteration function.

Applicants request review because the Examiner has not cited any part of the Andrei patent that discloses the identical invention claimed in claims 12, 25, and 38 of, under control of an iterator function processor, setting indicators to indicate which of the one or more predicates have been applied by the iterator function processor, and, under control of the data store engine, applying any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor.

Instead, the Andrei patent describes that engine 360 itself comprises parser 361, normalizer 363, compiler 365, execution unit 369, and access methods 370 (Col. 11, lines 5-7; Figure 3). The execution unit 369 generates calls into lower-level routines, such as the access methods 370, for retrieving relevant information (e.g., row 355) from the database table 350, and, after the plan has been executed by the execution unit, the server returns a query result or answer table back to the client(s) (Col. 11, lines 35-40). Thus, with the Andrei patent the execution unit itself executes the plan. Thus, the Andrei patent teaches away from, under control of an iterator function processor, setting indicators to indicate which of the one or more predicates have been applied by the iterator function processor, and, under control of the data store engine, applying any of the one or more of the predicates included in the statement that have not been applied by the iterator function processor.

Dated: May 8, 2007

By:__/Janaki K. Davda/_____

Janaki K. Davda
Registration No. 40,684

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984